Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims:

Claims 1-29 (canceled).

Claim 30 (new): A system for measuring the quality of a data network, the system comprising a client computer and a server computer, the client computer and the server computer being arranged to communicate via the data network using a connection-less transmission protocol, wherein:

the client computer is arranged to send data packets to the server computer so as to define transmitted data packets, each of the data packets having a first timestamp indicating a time at which said each data packet was sent to the server computer;

the server computer is arranged to:

receive at least one of the transmitted data packets so as to define received data packets and send, for each one of the received data packets, a corresponding modified data packet to the client computer so as to define modified data packets, the corresponding modified data packet having the first timestamp from said one received data packet and a second timestamp, the second timestamp indicating a time at which said one received data packet was received at the server computer; and

send a timeout data packet to the client computer when a predefined amount of time has elapsed since a most

recent one of the received data packets was received by the server computer; and

wherein the client computer is further arranged to:

store first and second log files with the modified data packets received from the server being stored in the first log file and the timeout data packet being stored in the second log file;

store each of the transmitted data packets in a ring buffer;

check each of the modified data packets received from the server with contents of the ring buffer; and

if one of the modified data packets is not received by the client computer within a length of the ring buffer, delete from the ring buffer a particular one of the transmitted data packets and which is associated with said one modified data packet and store the particular one transmitted data packet so deleted in the second log file.

Claim 31 (new): The system recited in claim 30 wherein:

the server computer comprises a server processor connected to a server memory, the server processor being arranged to read from the server memory a timeout value indicating a predefined amount of time, a server interval value and a quantity value; and

the server computer being further arranged to send successive timeout data packets, with an interval time equal to the server interval value, until a next one of the transmitted data packets is received or a maximum number of timeout data packets sent by the server computer equals the quantity value.

Claim 32 (new): The system recited in claim 31 wherein the client computer comprises a client processor connected to a client memory, the client processor being arranged to read from the client memory, for each one of the transmitted data packets, a client interval value, a client packet size value and a packet type value;

the client computer further being arranged to send the transmitted data packets with an interval time equal to the client interval value; and

each of the transmitted data packets having the packet type value and a client payload, the client payload filling said each transmitted data packet up to a first predefined size equal to the client packet size value.

Claim 33 (new): The system recited in claim 32 wherein the packet type value signifies either a "normal packet" or a "burst packet".

Claim 34 (new): The system recited in claim 32 wherein the client memory is a file stored on a disk.

Claim 35 (new): The system recited in claim 30 wherein each of the transmitted data packets comprises a sequence number.

Claim 36 (new): The system recited in claim 31 wherein the server processor is arranged to read, from the server memory, a server packet size value; and

the corresponding modified data packet having a server payload, the server payload filling the corresponding modified data packet up to a second predefined size equal to the server packet size value.

Claim 37 (new): The system recited in claim 32 wherein the client payload comprises a predefined or random bit pattern.

Claim 38 (new): The system recited in claim 36 wherein the server payload comprises a predefined or random bit pattern.

Claim 39 (new): The system recited in claim 30 wherein the connection-less transmission protocol is UDP/IP.

Claim 40 (new): The system recited in claim 31 wherein the client computer is further arranged to send a configuration packet to the server computer, the configuration packet comprising the timeout value, the server interval value, the quantity value and a server packet size value;

the server computer is further arranged to receive the configuration packet; and

and the server processor is arranged to write to the server memory the timeout value, the server interval value, the quantity value and the server packet size value from the configuration packet so received.

Claim 41 (new): The system recited in claim 31 wherein, for configuring the server computer, the client computer is further arranged to communicate with the server computer via the data network using a connection-oriented transmission protocol;

the server computer is further arranged to receive configuration data from the client computer, the configuration data comprising the timeout value, the server interval value, the quantity value and a server packet size value; and

and the server processor is arranged to write to the server memory the timeout value, the server interval value, the quantity value and the server packet size value from the configuration data so received.

Claim 42 (new): The system recited in claim 41 wherein the connection-oriented transmission protocol is TCP/IP.

Claim 43 (new): The system recited in claim 30 wherein the client computer is further arranged to calculate a round-trip time by subtracting the first timestamp from a timestamp indicating when the corresponding modified data packet is received in the client computer and store the round-trip time in the first log file.

Claim 44 (new): The system recited in claim 30 wherein the client computer is further arranged to calculate a client-to-server latency by subtracting the second timestamp from the first timestamp and store the client-to-server latency in the first log file.

Claim 45 (new): The system recited in claim 44 wherein the client computer is further arranged to read, from the first log file, a packet type value, the client-to-server latency and a round-trip time and calculate, for each one of the modified data packets having a packet type value equal signifying a "normal packet", a server-to-client latency by subtracting the client-to-server latency from the round-trip time.

Claim 46 (new): The system recited in claim 44 wherein the client computer is further arranged to read, from the first

log file, the packet type value and calculate a downlink bandwidth using the following equation:

(#BytesPerPacket+Overhead)*(#BurstPackets)/(t2-t1)

where:

#BytesPerPacket is, in bytes, a length of the modified data packet;

Overhead is, in bytes, a length of protocol header overhead;

#BurstPackets is a number of the modified data packets stored in the first log file having the packet type value signifying a "burst packet";

t1 is a timestamp indicating when a first one of the modified data packets is received in the client computer; and

t2 is a timestamp indicating when a last one of the modified data packets is received in the client computer.

Claim 47 (new): The system recited in claim 44 wherein the client computer is further arranged to read, from the first log file, the packet type value and calculate an uplink bandwidth using the following equation:

(#BytesPerClientPacket+Overhead)*(#BurstPackets)/
((t2+t4)-(t1+t3))

where:

#BytesPerClientPacket is, in bytes, a length of the data packet;

Overhead is, in bytes, a length of protocol header

overhead;

#BurstPackets is a number of the modified data packets stored in the first log file having the packet type value signifying a "burst packet";

t1 is a timestamp indicating when a first one of the modified data packets is received in the client computer;

t2 is a timestamp indicating when a last one of the modified data packets is received in the client computer;

t3 is the client-to-server latency for the first modified data packet; and

t4 is the client-to-server latency of the last modified data packet.

Claim 48 (new): The system recited in claim 44 wherein the client computer is further arranged to read, from the first log file, a round-trip time and a sequence number and determine a round-trip outage as follows:

if(Rtt2-Rtt1)>t5 then roundtrip outage=true

where:

Rtt2 is the round-trip time;

Rtt1 is the round-trip time of a previous one of the modified data packets, the client computer being arranged to identify the previous modified data packet by the sequence number; and

t5 is a predefined amount of time.

Claim 49 (new): The system recited in claim 44 wherein the client computer is further arranged to read, from the first

log file, the client-to-server latency and a sequence number and determine a downlink outage as follows:

if(Rtt4-Rtt3)>t5 then downlink outage=true

where:

Rtt4 is the client-to-server latency;

Rtt3 is the client-to-server latency of a previous one of the modified data packets, the client computer being arranged to identify the previous modified data packet by the sequence number; and

t5 is a predefined amount of time.

Claim 50 (new): The system recited in claim 45 wherein the client computer is further arranged to read, from the first log file, a sequence number and determine an uplink outage as follows:

if(Rtt6-Rtt5)>t5 then uplink outage=true

where:

Rtt6 is the server-to-client latency;

Rtt5 is the server-to-client latency of a previous one of the modified data packets, the client computer being arranged to identify the previous modified data packet by the sequence number; and

t5 is a predefined amount of time.

Claim 51 (new): The system recited in claim 48 wherein the client computer is further arranged to determine a number of lost packets between two of the modified data packets by

analyzing corresponding sequence numbers of the two modified data packets.

Claim 52 (new): The system recited in claim 43 wherein the client computer is further arranged to combine the round-trip time with low level measurement data from a mobile telephone or geographic information data.

Claim 53 (new): A method for measuring the quality of a data network, the system comprising a client computer and a server computer, the client computer and the server computer being arranged to communicate via the data network using a connection-less transmission protocol, the method comprising the steps of:

in the client computer, sending data packets to the server computer so as to define transmitted data packets, each of the data packets having a first timestamp indicating a time at which said each data packet was sent to the server computer;

in the server computer:

receiving at least one of the transmitted data packets so as to define received data packets and send, for each one of the received data packets, a corresponding modified data packet to the client computer so as to define modified data packets, the corresponding modified data packet having the first timestamp from said one received data packet and a second timestamp, the second timestamp indicating a time at which said one received data packet was received at the server computer; and

sending a timeout data packet to the client computer when a predefined amount of time has elapsed since

a most recent one of the received data packets was received by the server computer; and

further in the client computer:

storing first and second log files with the modified data packets received from the server being stored in the first log file and the timeout data packet being stored in the second log file;

storing each of the transmitted data packets in a ring buffer;

checking each of the modified data packets received from the server with contents of the ring buffer; and

if one of the modified data packets is not received by the client computer within a length of the ring buffer, deleting from the ring buffer a particular one of the transmitted data packets and which is associated with said one modified data packet and storing the particular one transmitted data packet so deleted in the second log file.

Claim 54 (new): The method recited in claim 53 further comprising the steps of:

reading from a server memory, in the server computer, a timeout value indicating a predefined amount of time, a server interval value and a quantity value; and

sending successive timeout data packets, with an interval time equal to the server interval value, until a next data packet is received or a maximum number of timeout data packets equals the quantity value.

Claim 55 (new): The method recited in claim 54 further comprising the steps of:

reading from client memory, in the client computer and for each of the transmitted data packets, a client interval value, a client packet size value and a packet type value; and

sending the transmitted data packets with an interval time equal to the client interval value, each of the transmitted data packets having a packet type value and a client payload, the client payload filling said each transmitted data packet up to a first predefined size equal to the client packet size value.